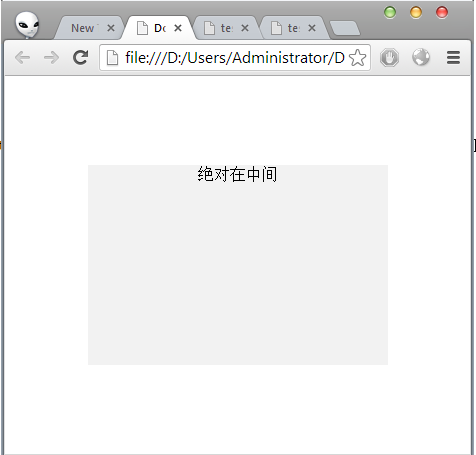
我的CSS笔记☹: (必须亲自实践, 必须配以插图)

**1.div居中问题:**

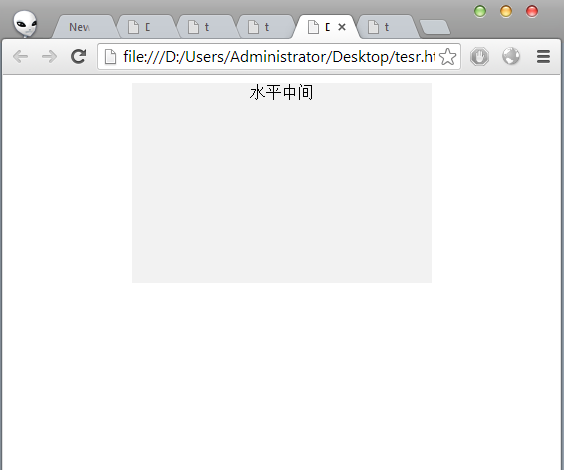
垂直水平居中:

|  |  |
| --- | --- |
| 1 | <!doctype html> |
| 2 | <htmllang="en"> |
| 3 | <head> |
| 4 | <metacharset="UTF-8"> |
| 5 | <title>Document</title> |
| 6 | <metahttp-equiv="X-UA-Compatible"content="IE=edge,chrome=1"> |
| 7 | <metaname="viewport"content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0"> |
| 8 | <scriptsrc='http://localhost/lib/jQuery/jquery-1.9.1.js'></script> |
| 9 | <style> |
| 10 | #center{ |
| 11 | *position*:absolute; |
| 12 | *width*:300px; |
| 13 | *height*:200px; |
| 14 | *top*:50%; |
| 15 | *left*:50%; |
| 16 | *background*: #f2f2f2; |
| 17 | *z-index*: 1; |
| 18 | *margin-left*: -150px; |
| 19 | *margin-top*: -100px; |
| 20 | *text-align*: center; |
| 21 | } |
| 22 | </style> |
| 23 | </head> |
| 24 | <body> |
| 25 | <divid="center">绝对在中间</div> |
| 26 | </body> |
| 27 | </html> |



水平居中:

|  |  |
| --- | --- |
| 9 | <style> |
| 10 | body{ |
| 11 | *text-align*: center; /\*for le IE8 父级元素内的内容居中\*/ |
| 12 | } |
| 13 | #center{ |
| 14 | *width*:300px; |
| 15 | *height*:200px; |
| 16 | *background*: #f2f2f2; |
| 17 | *text-align*: center; |
| 18 | *margin*: 0auto; |
| 19 | } |
| 20 | </style> |
| 21 | </head> |
| 22 | <body> |
| 23 | <divid="center">水平中间</div> |
| 24 | </body> |



**2有关footer的布局问题:**

<http://ryanfait.com/resources/footer-stick-to-bottom-of-page/>

<http://www.jb51.net/css/66892.html>

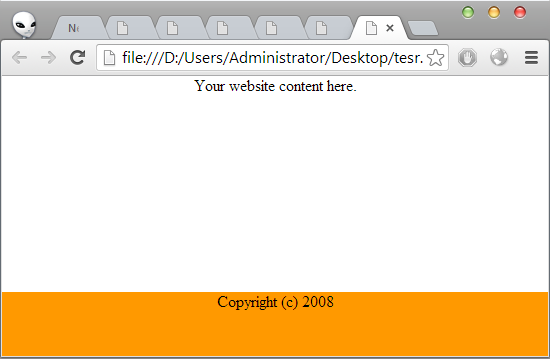
经常碰到body高度不够,footer半悬于页面之中,这样页面很不美观.

最初使用fixed来固定footer位置,很快就发现,当body内容超过屏幕高度时,fixed的footer就会遮住一部分内容,而且不能随着滚轮滚动.

后面google后发现可以使用负的margin值来实现,具体设置如下:

|  |  |
| --- | --- |
| 1 | <!doctype html> |
| 2 | <htmllang="en"> |
| 3 | <head> |
| 4 | <metacharset="UTF-8"> |
| 5 | <title>Document</title> |
| 6 | <metahttp-equiv="X-UA-Compatible"content="IE=edge,chrome=1"> |
| 7 | <metaname="viewport"content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0"> |
| 8 | <scriptsrc='http://localhost/lib/jQuery/jquery-1.9.1.js'></script> |
| 9 | <style> |
| 10 | \* { |
| 11 | *margin*: 0; |
| 12 | } |
| 13 | html, body { |
| 14 | *height*: 100%;   /\*必须要是100%,便于子元素设置百分比\*/ |
| 15 | *text-align*: center; |
| 16 | } |
| 17 | .wrapper { |
| 18 | *min-height*: 100%;        /\*设置其高度总是100%,即总是视口的高度,前提是需要其父元素body,html的高度为100%即视口的高度 IE6不支持min-height\*/ |
| 19 | *height*: auto!important;  /\*min-height hack for IE \*/ |
| 20 | *height*: 100%; |
| 21 | *margin*: 0auto-4em;  /\*这里设置的-4em就是你的footer的高度,是为了抵消当body高度高于视口高度时防止footer遮住一部分内容 push设置高度就用于此\*/ |
| 22 | *position*:relative;  /\*以便于里面的元素进行绝对定位后不会超过了wrapper容器\*/ |
| 23 | } |
| 24 | .footer, .push { |
| 25 | *height*: 4em; |
| 26 | *clear*:both;    /\*顺便还可以清除浮动.\*/ |
| 27 | } |
| 28 | .footer{ |
| 29 | *background*: #f90; |
| 30 | } |
| 31 | </style> |
| 32 | </head> |
| 33 | <body> |
| 34 | <divclass="wrapper"> |
| 35 | <p>Your website content here.</p> |
| 36 | <divclass="push"></div> |
| 37 | </div> |
| 38 | <divclass="footer"> |
| 39 | <p>Copyright (c) 2008</p> |
| 40 | </div> |
| 41 |  |
| 42 | </body> |
| 43 | </html> |

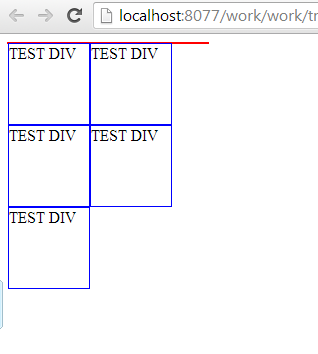
效果图:



**3.关于clearfix (清除浮动)**

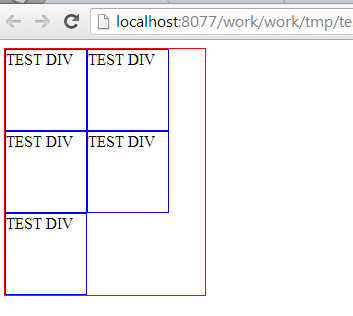
如果有一个DIV作为外部容器，内部的DIV如果设置了float样式，则外部的容器DIV因为内部没有clear，导致不能被撑开。

1. <div style="width:200px;border:1px solid red;">
2. <div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>
3. <div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>
4. <div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>
5. <div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>
6. <div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>
7. </div>



容器DIV没有被撑开  
大家可以看到，作为外部容器的边框为红色的DIV，没有被撑开。这是因为内部的DIV因为float:left之后，就丢失了clear:both和display:block的样式，所以外部的DIV不会被撑开。  
我们想让外部容器的DIV随着内部DIV增多而增加高度，要怎么解决呢？

以前我都是用这样的方法来解决：  
HTML4STRICT代码：

1. <div style="width:200px;border:1px solid red;">
2. <div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>
3. <div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>
4. <div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>
5. <div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>
6. <div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>
7. <div style="clear:both;"></div>
8. </div>  
   显示的结果如下：
9. 

显示正常了  
我们看到，在容器DIV内要显示出来的float:left的所有的DIV之后，我们添加了这样的一个DIV：<div style="clear:both"></div>。这样，其实就在最后增加了clear的动作。

或者使用overflow属性:

<div style="width:200px;border:1px solid red;**overflow:hidden**" class="clearfix">

<div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>

<div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>

<div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>

<div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>

<div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>

</div>

也可以清除浮动,使其父元素得到撑开.

有更好的方法来清除浮动:

1. .clearfix:after {
2. content: ".";
3. display: block;
4. height: 0;
5. clear: both;
6. visibility: hidden;
7. }

<div style="width:200px;border:1px solid red;" class="clearfix">

<div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>

<div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>

<div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>

<div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>

<div style="float:left;width:80px;height:80px;border:1px solid blue;">TEST DIV</div>

</div>

原来，这个clearfix的CSS使用了after这个伪对象，它将在应用clearfix的元素的结尾添加content中的内容。在这里添加了一个句号"."，并且把它的display设置成block；高度设为0；clear设为both；visibility设为隐藏。这样就达到了撑开容器的目的啦。

但是，在文章中说，Windows IE并不支持这样做。所以要让IE也完美显示，则必须在clearfix这个CSS定义的后面加上一些专门为IE设定的HACK。CSS如下：

CSS代码：

1. .clearfix:after {
2. content: ".";
3. display: block;
4. height: 0;
5. clear: both;
6. visibility: hidden;
7. }
8. /\* Hides from IE-mac \\*/
9. \* html .clearfix {height: 1%;}

/\* End hide from IE-mac \*/

综上: 关于子元素全都float后,父元素的高度collapse问题:

有4种解决方法:

1.使用一个子元素作为clear

2.父元素使用overflow:hidden (90%的情况下有效,当子元素的盒子的阴影值如果超过其高度,阴影值会被hidden)

3.父元素也使用float

4.现在有更好的clearfix: 如下:参考<http://nicolasgallagher.com/micro-clearfix-hack/>

.clearfix { zoom: 1; } /\* for IE \*/

.clearfix:before,

.clearfix:after { content: " "; display: table; }

.clearfix:after { clear: both; }

**4.关于css布局中的非常重要的vertical margin collapse问题:**

When two vertical margin touch,they collapse to the larger of the two value.

当两个垂直的margin接触时,此时的总margin值坍塌为其中的一个较大的margin值.

margin collapse 仅仅存在于垂直的margin中,水平的margin总是两个元素间的margin之和

注意:这里的margin touch是指的是两个垂直的元素间**没有任何的padding或者border**.正是因为垂直的元素间没有任何的padding或者broder,才造就了margin collapse 问题.

**5.fixed布局时注意的问题**

**他的参考点**是以浏览器的viewport参考的.

和absolute不同的是,fixed不以nearest的container的为参考点,fixed只和viewport有关

viewport即:你能看到的显示内容的浏览器的视窗宽度,它可以随浏览器的宽高度的变化而变化;

**6.relative属性的一般用途:**

1.对元素自己要进行较小的位置的改变时,比如移动文字位置.(因为**relative并没有使元素脱离文档流**,

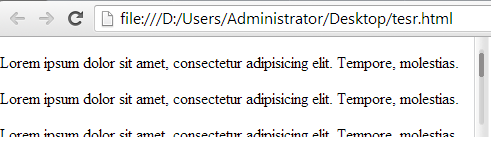
元素自己尽管有偏移,但是它还是占据着原来的位置,做较大的位置偏移当元素本身的空间变化时就会影响到周围的元素);

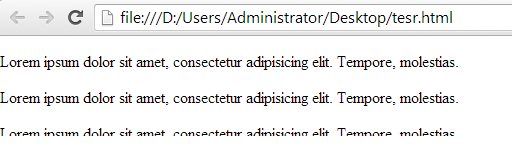
2.作为子元素的absolute的参考位置,此时只要设置属性为relative,不需要设置offset值

**7.使用CSS隐藏滚动条,但是扔然可以继续滚动的方法:**

**参考:**<http://hynchrstn.wordpress.com/2012/06/10/hide-scrollbar-but-still-scrollable-using-css/>需翻墙

|  |  |
| --- | --- |
| 1 | <!doctype html> |
| 2 | <htmllang="en"> |
| 3 | <head> |
| 4 | <metacharset="UTF-8"> |
| 5 | <title>Document</title> |
| 6 | <metahttp-equiv="X-UA-Compatible"content="IE=edge,chrome=1"> |
| 7 | <metaname="viewport"content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0"> |
| 8 | <scriptsrc='http://localhost/lib/jQuery/jquery-1.9.1.js'></script> |
| 9 | <style> |
| 10 | html, body { |
| 11 | *padding*: 0; |
| 12 | *margin*: 0; |
| 13 | *width*:200px; |
| 14 | *overflow*: hidden; /\*这里可以设置body的宽度,当超过body的宽度时就将其隐藏\*/ |
| 15 | /\*这里要注意下面的container已absolute,所以body(父元素还要设置position:relative属性)\*/ |
| 16 | /\*即可以将container上滚动条隐藏了\*/ |
| 17 | } |
| 18 | #container { |
| 19 | *position*: absolute; |
| 20 | *left*: 0; |
| 21 | *top*: 0; |
| 22 | *right*: -30px; |
| 23 | *bottom*: 0; |
| 24 | *padding-right*: 15px; |
| 25 | *overflow-y*: scroll; |
| 26 | *height*:100px; |
| 27 | } |
| 28 | </style> |
| 29 | </head> |
| 30 | <body> |
| 31 | <divid="container"> |
| 32 | <!-- lines, lines, lines --> |
| 33 | <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Tempore, molestias.</p> |
| 34 | <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Tempore, molestias.</p> |
| 35 | <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Tempore, molestias.</p> |
| 36 | <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Tempore, molestias.</p> |
| 37 | <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Tempore, molestias.</p> |
| 38 | <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Tempore, molestias.</p> |
| 39 | <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Tempore, molestias.</p> |
| 40 | </div> |
| 41 | </body> |
| 42 |  |
| 43 | </html> |





主要是通过:定位来隐藏其滚动条.即将滚动条使用绝对定位移出内容的宽度.

**8.关于block元素和inline元素的区别:**

对于行内元素(inline elements)的特点即display:inline;在CSS样式中想要设置它的高度(height)和宽度(width)会发现没有效果.

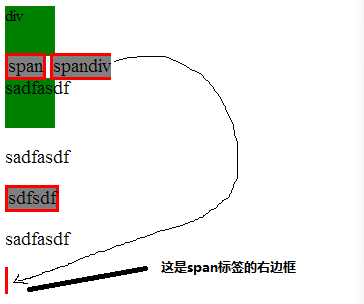
行内元素的height和width是随着和它包含的内容变化的.

对于块级元素的特点:可以自由的设置其高度(heighe)和宽度(width),同时自动换行.

通常情况下:对于页面大的区块使用<div>标签,而<span>标记仅用于需要单独设置样式风格的小元素,例如一个单词,一幅图片,一个超链接等.

注意:<span>标签内最好不要包含<div>标签或者<p>标签即有display:block属性的元素,即使包含,如下的奇怪错误:

|  |  |
| --- | --- |
| 1 | <!doctype html> |
| 2 | <html> |
| 3 | <head> |
| 4 | <metahttp-equiv="content-type"content="text/html; charset=utf-8" /> |
| 5 | <style> |
| 6 | div{ |
| 7 | *background-color*:green; |
| 8 | /\*display:inline;\*/ |
| 9 | *height*:50px; |
| 10 | *width*:50px; |
| 11 | } |
| 12 | span{ |
| 13 | *background-color*:gray; |
| 14 | /\*display:block;\*/ |
| 15 | *height*:100px; //由于span是inline元素,这里的高度宽度会不起作用 |
| 16 | *width*:100px; |
| 17 | *font-size*:19px; |
| 18 | *border*:3pxsolidred; |
| 19 | } |
| 20 | </style> |
| 21 | </head> |
| 22 | <body> |
| 23 | <div>div</div> |
| 24 | <span>span</span> |
| 25 | <span> |
| 26 | spandiv |
| 27 | <div>sadfasdf</div> |
| 28 | <p>sadfasdf</p> |
| 29 | <span>sdfsdf</span> |
| 30 | <p>sadfasdf</p> |
| 31 | </span> |
| 32 | </body> |
| 33 | </html> |



**9.关于css的注释符:**

注意这里我使用的是单行注释符// ,但是在CSS中这种注释符对不同的浏览器是不同的

比如IE10和IE9,IE8其中IE10支持单行注释,但是IE9,8 就不支持,当然chrome和firefox

是支持的,这里为我为了简单就使用单行注释符//

多行注释符才是CSS的通用注释符**最好不要使用单行注释符,为了浏览器的兼容性**

**10.关于position定位:**

position定位:

position顾名思义:位置,即指定块的位置.

注意position指定块的位置的参考系:

①自己相对于父块的位置 -🡪absolute定位

②自己相对于自己原先在文档流中的位置的偏移🡪relative定位

position有4个属性值:

static, absolute, relative, fixed.

其中static为其默认值,指的是块原本应该在的位置.即该值没有任何的移动效果

绝对定位是参照浏览器的左上角,配合top,ritght,bottom,left进行定位.

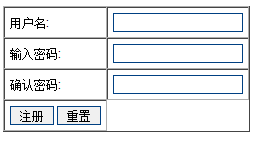
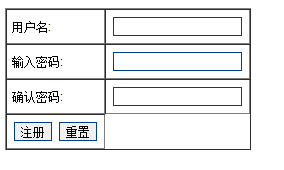
**它默认是依照父级的坐标原点为原点的.**

But,如果有设定了top,right,bottom,left,并且父级没有设定position属性,那么此时的absolute的参考点则以浏览器的左上角为原点进行定位,位置将由这4个属性决定.

如果父级有设置参考点那么就是以**父级的坐标原点为原点(父级的左上角点)为参考点**

对于Z-index的空间定位前提:必须设置position属性为absolute即脱离文档流.

**11.关于表格的一些CSS属性(消除边框的重叠)**



(chrome显示效果) (firefox显示效果)

<table cellpadding="5" cellspacing="0" border="1">

<tr><td>用户名:</td><td><input type="text" name="User"></td></tr>

<tr><td>输入密码:</td><td><input type="text" name="passwd1"></td></tr>

<tr><td>确认密码:</td><td><input type="text" name="passwd2"></td></tr>

<tr>

<td colspan="1" align="center">

<input type="submit" value="注册">

<input type="reset" value="重置">

</td>

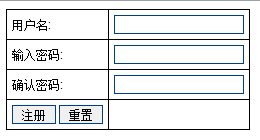
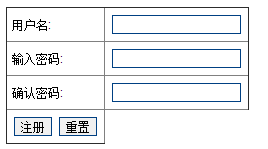
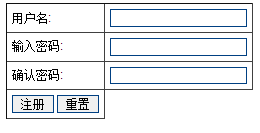
</tr>

</table>

会发现当border="1"时有边框重叠的现象:

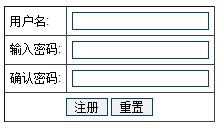
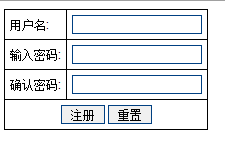
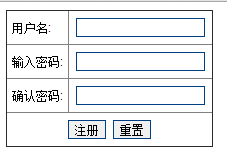
使用样式border-collapse:collapse; /\*消除边框的重叠\*/

此时

(chrome显示效果) (firefox显示效果) (IE显示效果)

那这里的colspan="1"有什么作用呢?当把其值改为colspan="**2**"时有如下效果:



此时colspan就是可以跨列显示

至于cellpadding="5" cellspacing="0"改变其一下数值即可知道.

**12.一些常用的css hack**

3像素问题及解决办法

当使用float浮动容器后，在IE6下会产生3px的空隙，有意思的是右侧容器没设置高度时3px在右侧容器内部，当设定高度后又跑到容器的左侧了，所以对布局精度要求高的话，请参考例29、31的解决方法

上边容器浮动后，下边的容器跟着浮动，造成页面错乱

如以上例子中的footer必须要单独占一行，但当sidebar浮动后，content的高度小于 sidebar的高度时，footer也跟着浮动到sidebar右侧了，造成页面错乱。解决办法是在maincontent和footer之间插入一个容器，设置样式 clear:both; height:0; font-size: 1px; line-height: 0px;用clear:both清除，即可使页面正常

**当子元素浮动且未知高度时，怎么使父容器适应子元素的高度？**

这种情况可在父窗口加上 overflow:auto;zoom:1;这两个样式属性，**overflow:auto;是让父容器来自适应内部容器的高度**，zoom:1;是为了兼容 IE6而使用的CSS HACK。zoom:1;通不过W3C的验证，这也是遗憾的一点，幸好IE支持<!--[if IE]>这种写法，可以专门针对IE来写单独的样式，所以可以把这个属性写在页面内的<!--[if IE]>中，这样应该可以通过验证了

相对定位与绝对定位

规划页面时，许多时候需要用到相对定位或绝对定位，这里边有个技巧，掌握后有些地方就不用繁冗的js了。就是当父容器使用相对定位后，此时子元素再使用绝对定位，位置是相对于父元素。如果父元素没有定位，那么子元素的绝对定位是相对于浏览器

IE6双倍边距bug

当页面内有多个连续浮动时，如本页的图标列表是采用左浮动，此时设置li的左侧margin值时，在最左侧呈现双倍情况。如外边距设置为10px，而左侧则呈现出20px，解决它的方法是在浮动元素上加上display:inline;的样式，这样就可避免双倍边距bug

**RD8010: IE6 IE7(Q) IE8(Q) 不支持 'min-width' 和 'max-width' 特性**

i**e8中图片设置maxwidth属性满足一定的条件会导致消失**

<http://www.yuanchuang.net/content/14132.html>

<http://www.w3help.org/zh-cn/causes/RD8010\>

当img: max-width:100%

